DOCUMENT RESUME

ED 047 379 EA 003 289

AUTHOR Etnyre, Vance A.
TITLE GENIRAS: A Class of General Information Retrieval and Application Systems.
PUB DATE 20 Oct 70
NOTE 12p.; Paper presented at Institute of Management Sciences American Meeting (11th, Los Angeles, California, October 19-21, 1970)

EDRS PRICE EDRS Price MF-$0.65 HC-$3.29
DESCRIPTORS *Computer Oriented Programs, Data Bases, *Data Processing, Information Retrieval, Information Storage, *Information Systems, Management Systems

ABSTRACT

General Information Storage, Retrieval, and Application System (GENIRAS) is a computer-aided information system designed for the storage, retrieval, and manipulation of data files. The system gives the user (1) flexibility in defining and using data bases, (2) a wide range of procedures to be applied to the data, and (3) a variety of acceptable programing languages for data analysis. (RA)

EDO 47379

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

GENIRAS:

A CLASS OF GENERAL

INFORMATION RETRIEVAL AND

APPLICATION SYSTEMS[1]

BY

VANCE A. ETNYRE[2]

UNIVERSITY OF ILLINOIS, URBANA, ILLINOIS

ED 003 289

[1]Presented at the 11th American Meeting of the Institute of Management Sciences in Los Angeles, California, October 19-21, 1970.

[2]Lecturer, Department of Business Administration and Administrative Associate, College of Engineering.

GENIRAS is the name given to a class of computer-aided information systems designed for the storage, retrieval, analysis, manipulation, and display of various files, records, and associated data items. The acronym GENIRAS (pronounced as "generous") is derived from the words GENeral Information Storage, Retrieval and Application System. The name is intended to characterize the integration of the storage and retrieval functions used in most generalized file management systems with the application functions called for in "management information systems."

GENIRAS systems provide mechanisms for interaction at various levels with computer-based information systems. At the most primitive level, standard procedures may be applied to pre-defined data bases using a simple "English-like" language. At a more sophisticated level, a new set of procedures may be defined and applied to new data bases. This may be accomplished by using a simple "English-like" language provided by the system, the full power of a procedural language, assembly language, or various combinations of these. GENIRAS systems also allow any user to define his own special purpose language and use this new language for further processing.

Several assumptions lead to the definition and development of this class of systems. Included in these assumptions were the following:

(1) Most data are recorded in meaningful groups or structures rather than in random pieces. A "data group" would be that collection of data items which share one or more common attributes. An attribute might describe a common organizational boundary, a specific time period, or a specific functional relationship.

(2) Most data processing applications require that one or more procedures be executed for every member of a specific data group. The process of assuring that every member of a data group is considered within a certain procedure is called "exhausting" the data group.

(3) Many data processing applications require that a given procedure or set of procedures be applied to every set of "corresponding elements" from various data groups. The term "corresponding elements" means elements which share one or more specified attributes. The process of finding corresponding elements is usually called "matching."

(4) Many individuals understand their data processing needs well enough to describe them easily and precisely to a friend or colleague. These individuals often are unable to use the power of a computer to help solve their problems because languages available for use on the computer are completely different from the language used to instruct a colleague.

(5) The designer of a general information system cannot predict in advance every specific task that his system will be required to perform.

(6) Often a "new" data processing application requires that a standard processing technique be applied to a data group or data file which had not previously undergone that particular process.

GENIRAS systems give the user a great deal of ease, power and flexibility in defining and using (1) data bases, (2) procedures to be applied to data bases, and (3) languages to be used in applying procedures to data bases.

In many cases, the standard procedures supplied by GENIRAS systems can be applied directly to a user's data to satisfy simple requests. This often requires only one or two simple statements in a user-oriented language to connect the specified action with the selected data. GENIRAS systems take advantage of the structure of data groups to relieve the user of the need to match corresponding items and exhaust data groups. These functions are handled automatically within a GENIRAS system. Procedures for copying, sorting, updating and transforming data files also are supplied as a convenience to the user.

Definition of Data Bases

GENIRAS systems allow users to access various data bases with great ease. If a suitable data base has already been defined, the user needs only to call for it by name. Otherwise, the user may define his own data base.

Data bases are defined by specifying and describing the files, subfiles, records, and items to be included within each data base. Each file within a data base must be given a name. If a file is to be separated into several subfiles, then each subfile must also be given a unique name. Another name can be used to describe the records which occur within a particular file or subfile. Data items (fields within the records) are defined by giving the name, data type, length, and starting location of each field to be included within the data base. Once a data item has been defined, only its name is needed for references within a GENIRAS system. The following example shows how part of a data base might be defined.[1]

```
DEFINE DATA.
*BEGIN
FILE=PAYROLL    RECORD=PAY_RECORD
SOC_SEC_NO=A(9),1
NAME=A(20),10    DEPT=A(5),31
TITLE=A(12),40    WAGE_RATE=N(6.2),55
   .
   .
   .
   .
*END
```

In naming data items, the user may use any combination of letters, numbers, and special symbols, but must be sure that the names are unique. If blanks or special symbols are used within a data name, the entire name must be included within single quote marks each time it is referenced. Thus, the following may be used as data names in a GENIRAS system:

```
OVERTIME_WAGES
'OVERTIME WAGES FOR NON-PERMANENT WORKERS'
'OVERTIME WAGES'
'OVERTIME-WAGES'
'O.W./N.P.W.'
OVRTMWGS
'OW*+@ 2Z:#'
```

Several names may be assigned to the same data item, so that full descriptive names may be used when clarity is needed, and shorter names may be used when speed and convenience are desired. Names may be added or changed at any time.

If the same name was given to items from two different files, then both the item name and the file name must be used to uniquely identify the data item. When a name is used within several subfiles, both the file and subfile names must be added to the item name to uniquely identify each item. Using this scheme, the following names can be used to identify data items:

```
DEPARTMENT:PAYROLL
DEPARTMENT:TRANSACTIONS
DATE:PAYROLL:OLD
DATE:PAYROLL:CURRENT
NAME:FILE4
NAME:FILE6:SUBFILE3
```

Definition of Procedures

To allow GENIRAS systems sufficient flexibility to meet the needs of various users, the procedures used by a GENIRAS system may come from any of the following sources: (1) the system designer, (2) the system implementor, (3) the individual users of the system.

General purpose procedures which are needed to support a wide spectrum of activities are provided by the designer of the GENIRAS system. Included in this category are: retrieval and selection routines, sorting routines, updating and copying routines, tabular and graphical display routines, arithmetic operations, and some forms of statistical analysis.

At the most primitive level, these routines are executed using statements in a simple, English-like language. For example:

```
COPY TRANSACTIONS ONTO TAPE1.
LIST NAME,AGE,AND SALARY FOR DEPT(40).
GRAPH SALARY_TOTALS AND EXPENSE_TOTALS AGAINST YEAR FOR DEPT(40).
SORT SALARY_FILE; CONTROL='DEPARTMENT,NAME'.
```

At a more sophisticated level, standard system functions may be called as subroutines from a user's procedural language or assembly language program.

Special Purpose Routines

Special purpose procedures can be added to a GENIRAS system by the system's implementor--the person or persons responsible for maintaining the system at a particular installation. These special procedures might include routines for editing or updating special files, special forecasting procedures, or display programs for specific reports. These routines can be written in a procedural language, assembly language, or a combination of these.

Suppose, for example, that procedures named TRANSACTION_REPORT, INVENTORY_REPORT, and CUMMULATIVE_UPDATE have been added to a GENIRAS system

by the system implementor or system manager. At the most primitive level of usage, these procedures can be executed by entering the following statements:

```
EXECUTE CUMMULATIVE UPDATE USING MASTER_1 AND NEW_FILE.
PRINT TRANSACTION_REPORT FOR DEPT(22).
SORT INVENTORY_FILE;  CONTROL=DEPT.
PRINT INVENTORY_REPORT; CONTROL=DEPT.
```

These procedures may also be executed as subroutines from a user's procedural language or assembly language program.

User Defined Procedures

There are two ways in which a user may define new procedures within a GENIRAS system. If the user only needs to combine standard system procedures in a particular way, he can use the DEFINE feature of the GENIRAS system. An example of such a usage might be:

```
DEFINE 'PERCENTAGE LIST'  (PROCEDURE) USING A AND B.
*BEGIN
ADD A TO B GIVING C.
DIVIDE C BY A GIVING RATIO.
MULTIPLY RATIO BY 100 GIVING PERCENT.
LIST A, B, C, AND PERCENT.
*END
```

To execute a procedure which has been defined in this way, the user submits a simple PERFORM statement containing the name of the procedure and the names of all arguments to be used with the procedure. To execute the procedure 'PERCENTAGE LIST' (defined in last example) the following statement might be submitted:

```
PERFORM 'PERCENTAGE LIST' USING SALARIES(1969) AND SALARIES(1970).
```

The procedure 'PERCENTAGE LIST' was defined using the "dummy" arguments A and B. When the procedure is executed, the names of the real arguments would be taken from the PERFORM statement. Thus, the items identified by SALARIES(1969) and SALARIES(1970) would be used in the percentage calculation and the listing. The user could also submit the statement:

PERFORM 'PERCENTAGE LIST' USING EXPENSES(1970) AND EQUIPMENT_COST(1970).

and the same procedure would be applied to different sets of data items.

A user of a GENIRAS system may also define new procedures by writing a procedural language program (PL/1, FORTRAN, COBOL) and using the COMPILE feature of the GENIRAS system. Assume, for example, that a program called PROGRAM1 has been coded in the PL/1 programming language, and this source program has been stored in a file called PROGFILE. The statement

```
COMPILE PROGRAM1(PL1) FROM PROGFILE.
```

would be sufficient to compile the source (PROGRAM1) and store the result in a temporary users' library.

If a COBOL program named PROGRAM2 was submitted as a card deck, and the object program was to be stored in a users' library called LIBRARY2, the statements

```
COMPILE PROGRAM2(COBOL) INTO LIBRARY2.
*BEGIN
 (card deck for PROGRAM2)
*END
```

would accomplish the task. These programs could be executed using the following statements:

```
EXECUTE PROGRAM1.
EXECUTE PROGRAM2 FROM LIBRARY2.
```

Selection Variables

GENIRAS systems allow users to reference entire data groups (all items having the same name) or subgroups (only those items with the specified name which have certain selection attributes). The variables which determine the selection attributes of an item are designated when the item is defined. For example, if YEAR was designated as a selection variable for the item named

EXPENSES, then the word EXPENSES would refer to an entire data group and the terms EXPENSES(1968) and EXPENSES(1970) would refer to subgroups contained within the data group called EXPENSES. Using this technique, graphs of EXPENSES by department for the last three years could be obtained with the statements:

```
GRAPH EXPENSES(1968) BY DEPARTMENT.
GRAPH EXPENSES(1969) BY DEPARTMENT.
GRAPH EXPENSES(1970) BY DEPARTMENT.
```

Selection Criteria

Special criteria may also be used within a GENIRAS system for the selection of particular sets of data items. Such criteria are defined by using names within a known data base or column numbers within a record. Each criterion is given a name which may be used each time the criterion is to be applied. For example, a criterion named 'LAST 5 YEARS-NO EQUIP' could be defined using the statement:

```
DEFINE 'LAST 5 YEARS-NO EQUIP' (CRITERION).
*BEGIN
YEAR>1965 AND YEAR<1970
AND NEW-EQUIPMENT=0
*END
```

This criterion could be applied to one or more procedures using statements like the following:

```
LIST TOTAL EXPENSES BY DEPARTMENT AND YEAR;
   CRIT='LAST 5 YEARS-NO EQUIP'.

PRINT EQUIPMENT_REPORT(CRIT='LAST 5 YEARS-NO EQUIP').
```

Modes of Interaction with the System

GENIRAS systems have been designed to operate in interactive mode, remote batch mode, or local batch mode. Interactive mode of operation requires that a processor is available on a time-slicing or direct interrupt basis. Several

versions of GENIRAS systems have been tested at the University of Illinois using local batch environment with punched card input, and remote batch environment using teletypewriters for input.

There have been several forms of input languages supported by GENIRAS systems. The first language to be implemented consisted of a series of simple assignment statements. To use the system, one would enter a statement such as:

```
VERB='LIST', FIRST_NAME='SALARY COSTS',
SECOND_NAME='EQUIPMENT COSTS'; FIRST_INDEX='1970';.
```

Later versions supported a simple "English-like" language called GENIRAL (for GENeral Information Retrieval and Application Language). The examples given earlier in this paper were written in GENIRAL.

The Version 3 GENIRAS system at the University of Illinois allows any user to define his own language for interaction with the GENIRAS system. Once the syntax of the language has been defined, statements within this language will be interpreted and control will be passed to whatever procedures are required to execute the statements. This feature[2] allows the user to define languages for "free form" data input, or languages specifically adapted to certain types of problems (accounting problems or network problems for example).

Summary

GENIRAS systems give the user a great deal of ease, power and flexibility in defining and using (1) data bases, (2) procedures to be applied to data bases, and (3) languages to be used in applying procedures to data bases.

In many cases, the standard procedures supplied by GENIRAS systems can be applied directly to a user's data to satisfy simple requests. This often requires only one or two simple statements in a user-oriented language to connect the required action with the selected data. GENIRAS systems can utilize the structure of data groups to automatically match corresponding items and exhaust data groups.

For specialized applications, the user may define his own data bases and his own procedures to operate on the data. He may also define his own language for a special type of application.

FOOTNOTES

[1] In this example, the numbers inside of parentheses give the lengths of the fields being defined and the numbers which follow the parentheses give the starting location of the field within PAY_RECORD.

[2] Based on the principles of the Translator Generator, developed by Professors S. J. Fenves and L. E. Murphree at the University of Illinois, Civil Engineering Systems Laboratory.